An aspect query language model based on query decomposition and high-order contextual term associations

Journal Item

An Aspect Query Language Model Based on Query Decomposition and High-Order Contextual Term Associations

Dawei Song and Qiang Huang

School of Computing

The Robert Gordon University

St Andrew Street, Aberdeen, AB25 1HG, United Kingdom

{d.song, q.huang}@rgu.ac.uk

Peter Bruza

Faculty of Information Technology

Queensland University of Technology

GPO Box 2434, Brisbane QLD 4001, Australia

{p.bruza}@qut.edu.au

Raymond Lau

Department of Information Systems

City University of Hong Kong

Tat Chee Avenue, Kowloon, Hong Kong SAR

{raylau}@cityu.edu.hk

Received: date / Accepted: date

Abstract

In information retrieval research, more and more focus has been placed on optimizing a query language model by detecting and estimating the dependencies between the query and the observed terms occurring in the selected relevance feedback documents. In this paper, we propose a novel Aspect Language Modelling framework featuring term association acquisition, document segmentation, query decomposition, and an Aspect Model for parameter optimization. Through the proposed framework, we advance the theory and practice of applying high-order and context-sensitive term relationships to information retrieval (IR). We first decompose a query into subsets of query terms. Then we segment the relevance feedback documents into chunks using multiple sliding windows. Finally we discover the higher order term associations, i.e., the terms in these chunks with high degree of association to the subsets of the query. In this process, we adopt an approach by combining the Aspect Model (AM) with the Association Rule (AR) mining. In our approach, the AM not only considers the subsets of a query as "hidden" states and estimates their prior distributions, but also evaluates the dependencies between the subsets of a query and the observed terms extracted from the chunks of a feedback document. The AR provides a reasonable initial estimation of the high-order term associations by discovering the associated rules from the document chunks. Experimental results on various TREC collections verify the effectiveness of our approach, which significantly outperforms a baseline language model and two state-of-the-art query language models namely the Relevance Model and the Information Flow model.

Keywords: Information Retrieval, Association Rules, Aspect Model, Query Decomposition, Document Segmentation

1 Introduction

It is generally acknowledged that Information Retrieval (IR) is a context-sensitive task and it is equally acknowledged that it is a significant challenge to make IR systems sensitive to context. By way of illustration, given a query "Java", an IR system may return documents about "programming" and others about "Merapi" (a volcano on central Java Island), as they all contain the term "Java". If the retrieval context is information technology, documents about "programming" are relevant. However, for a volcanologist, documents about "Merapi" are more likely to be relevant. The field of IR has developed a number of techniques to deal with queries like the one just presented. For example, interactive query reformulation may offer alternatives such as "java programming" to the user from which to select in order to disambiguate the query. Techniques such as pseudo-relevance feedback rely on the underlying corpus to automatically expand the query with associations into one which will hopefully better align with the user's given retrieval context. Recent work in this vein has attempted to model the retrieval context by the so called higher-order associations such as "Java, computer $\rightarrow$ programming" and "Java, volcanologist $\rightarrow$ Merapi" (Lau et al. 2008). The syntax of these associations echoes earlier work motivated in logic-based IR in which the symbol "$\rightarrow$" can loosely be interpreted as an implication relation. It should be stressed this implication is not based on truth-values, but attempts to reflect relevant associations which may be triggered in human memory. In other words, the model theory underpinning higher-order associations is not logical, but ultimately cognitive. This distinction is easy to state, but a hard one to drive home in practice. We will not develop it further in this account but will be echoed later in the choice of knowledge representation.

The promise of strong high-order associations is that they may provide a basis for IR systems which are more sensitive to context. For pragmatic reasons in

their original conception, the terms in the premise of a given higher order association need not be a valid phrase (Song and Bruza 2003). Therefore, high-order term associations would seem to be an expressive and pragmatically effective means of representing associations which may be automatically captured in a pseudo-relevance feedback setting, or via implicit relevance feedback e.g., click-through information (Shen et al. 2005), or within a range of other contextual factors such as time, location, task at hand, etc. (Ingwersen and Belkin 2004; Ingwersen and Jarvelin 2005; Ingwersen 2001; Jones and Brown 2004).

The objective of this article is to exploit higher-order associations in order to produce more effective query models. Query models computed from the statistical language modelling framework provide both a sound theoretical basis as well as encouraging improvement in retrieval effectiveness (Lafferty and Zhai 2001; Lavrenko and Croft 2001; Song and Bruza 2003; Bai et al. 2005; Cao et al. 2005). In query language modelling, documents and queries are represented as language models (probability distributions over a vocabulary of terms). The matching process involves a measure of "distance" between two language models, i.e., the query model M_Q of a query Q and the document model M_D of a document D respectively. The smaller the distance is, the more similar the two models are, and hence, the more likely it is for D to be relevant to Q. A typical distance measure is the Kullback-Leibler (KL) divergence:

$$Dist_{KL}(M_Q||M_D) = \sum_i PQ_{q_i} \log \frac{PQ_{q_i}}{PD_{q_i}} \tag{1}$$

In practice, query language modelling (QLM) boils down to a query expansion process through relevance feedback. The key question is how to derive an accurate query model which aligns with the user's retrieval context.

Classical LM approaches (Ponte and Croft 1998) make use of uni-grams or bi-grams to build a language model. Many approaches exploit relevance feedback

documents to compute a query model (Lafferty and Zhai 2001; Lavrenko and Croft 2001; Zhai and Lafferty 2001). One example is the Relevance Model (RM) (Lavrenko and Croft 2001), which estimates the joint probability of observing a term w in the vocabulary together with query topic $Q = \{q_1, \cdots, q_{|Q|}\}$. The assumption of independence among query terms has been made to reduce the complexity of computation. This, however, neglects the relationships between terms in determining the query language model and therefore may lead to inappropriately high probabilities being ascribed to terms which are not aligned with the given retrieval context. In response to this, more recent research proposes query language models which are more sensitive to term relationships or dependencies (Song and Bruza 2003; Pickens and MacFarlane 2006; Bai et al. 2005; Cao et al. 2005; Metzler and Croft 2007), for example, grammatical links (Gao et al. 2004), or term co-occurrence and WordNet relations (Cao et al. 2005).

In the wake of this line of research, there has been a trend of decomposing a query into different combinations (subsets) of query terms, and exploiting term relationships derived from the subsets of query terms rather than traditional pairwise term co-occurrences. For example, the initial query "Java, volcanologist" can be decomposed into "Java", "volcanologist", "Java, volcanologist". On one hand, intra-query dependency is taken into account. On the other hand, different aspects (in the form of query term subsets) of the query are also considered to establish the overall association between the initial query and a potential expansion term. Song and Bruza (2003) propose an information flow model to explicitly capture the high-order term relationships, and in (Pickens and MacFarlane 2006), the authors build a term context model based on a maximum entropy algorithm to estimate the co-occurrence of terms in documents with the query topic. Metzler and Croft (2007) expand the approach used in (Pickens and MacFarlane 2006), and decompose the query topic into "latent"

concepts, which consist of the combinations of query terms. However, no explicit high-order term relationships were used.

In (Bai et al. 2005), which is more significantly related to this paper, high-order inferential term relationships extracted by the information flow approach (Song and Bruza 2003) have been employed in a LM framework combining the effects of information flows from different subsets of query terms. Essentially, the Information Flow approach (Song and Bruza 2003) is based on a lexical semantic space model, namely Hyperspace Analogue to Language (HAL). The HAL space is constructed by moving a fixed length sliding window over the corpus by a one term increment. All terms within the window are considered as co-occurring with each other with strengths inversely proportional to the distance between them. After traversing the corpus, numeric vectors representing the concepts (terms) are produced. Arbitrary terms (e.g., "Java" and "computer") that are related to each other (but not necessarily the syntactically valid phrases) can be combined to form a new concept, also represented as a vector, by a weighted addition of the underlying vectors of the terms. The information flow between two concepts is then computed by measuring the degree of inclusion between their underlying vectors.

Despite its good performance, re-loading and manipulating vectors in the pre-computed HAL space, which is normally very large, for each query session may potentially lead to a high computational overhead. In particular, for query decomposition, the expensive information flow computation process (sequential scan of the vocabulary to compare each vector in the HAL space with the vector representing a subset of query terms) has to be performed for $2^{|Q|}$ times, i.e., for each of the subsets of query terms. Indeed, as a consequence, in both (Song and Bruza 2003) and (Bai et al. 2005) the query decomposition was not actually performed. It was instead approximated by computing information flows only

once from the whole set of query terms only. Furthermore, the fixed sized sliding window approach used in HAL is less flexible to encode various levels of associations between terms, and different segments in the documents as well as different query term subsets should not be treated equally in generating the high-order term associations.

1.1 Our Proposed Approach

This paper aims to further advance the trend of using query decomposition by incorporating high-order term relationships. A query is first decomposed into subsets of query terms, then estimates are computed reflecting the dependencies between each query subset and the observed words in the (pseudo-) relevant documents. In order to improve the accuracy of the estimation procedure, in this paper, we propose a novel framework including the following key features.

1. The use of association rule mining from documents, which is able to capture high-order term associations from all different subsets of query terms in one go, thus truly realizing the idea of query decomposition.

   Association rules originally aim to capture the association patterns between items in a transaction database in an almost identical form as the high-order term associations, e.g., "Java, volcanologist $\Rightarrow$ Merapi". The mining of association rules has been widely used not only in Web usage pattern analysis (Srivastava et al. 2000), intrusion detection (Luo and Bridges 2000) and bioinformatics (Creighton and Hanash 2003), but also in the text-based knowledge discovery. Efficient and effective algorithms are also developed to further improve the performance of the association rule based systems. More details will be given in Section 3.2.

2. Dividing the documents into variable length segments through multiple sliding windows of different sizes to perform association rule mining.

Using shorter segments instead of the whole documents reduces the computational load of association rule mining. On the other hand, using viable length windows for document segmentation enables different levels of term associations generated from different sized segments to be taken into account in a mixture model. To our knowledge, there has not been an approach to the use of multiple-sized sliding windows for query language modelling.

3. A so-called Aspect Model to establish a new query language model by aggregating the high-order term associations between the different query subsets and the observed terms in documents, and optimize the prior probabilities of query subsets and document segments.

   In the model, by treating the query subsets as different aspects of the query, the underlying idea is to view the query from different angles and focal points in order to get a holistic view as well as to examine the specific aspects of the query. The document segments and terms are connected through one or more query subsets (aspects), with an EM algorithm to estimate the parameters involved, e.g., different segments are associated with different weights in relation to a query subset. The details about the model and the automated parameter optimization algorithm will be presented in the next sections.

In short, this article proposes a novel framework integrating the advantages of association rule mining, multiple window segmentation, query decomposition, and statistical learning for parameter optimization to derive higher-order term relationships for incorporation in query language models. An extensive empirical evaluation demonstrates a superior performance of our approach in comparison with a baseline language model and two existing high-performance query expansion models: the Relevance Model and the Information Flow model.

2 Aspect Query Model

This section presents relevant underlying theory, in particular, how a query language model is constructed by integrating the contributions from the decomposed subsets of a query and the segments of a document.

2.1 Basic Structure

Let $Q = \{q_1, \cdots, q_{|Q|}\}$ be a query, where q_j stands for single query term and $|Q|$ is the length of Q. We first decompose the query into subsets of query terms. Consequently, an example query $Q = \{q_1, q_2\}$ can be transformed into $Q' = \{\{q_1\}, \{q_2\}, \{q_1, q_2\}\}$, as further illustrated in Fig. 1. Let Q_j denote one of the query subsets.

$$\begin{array}{c} Query\ \ (Q) \quad \rightarrow \quad decomposed\ Query\ \ (Q') \\ \{theory, derivation\} \ \rightarrow \ \{\{theory\}, \{derivation\}, \{theory, derivation\}\} \end{array}$$

Figure 1: Example of the query decomposition

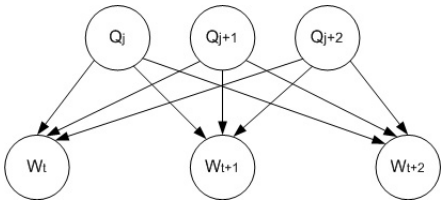


Figure 2: A graphic model of the relations between the subset of query Q_j and the observed word w_t

After query decomposition, a mechanism is needed to derive query language model $P(w|Q)$ by taking into account the high-order associations between Q_j and each observed word w in the (pseudo-) relevant documents. Fig. 2 shows a graphic model, in which each subset of query terms, Q_j, can have an impact on w. Here, we integrate all the possible contributions from each Q_j to derive the following equation:

$$P(w|Q) = \sum_{Q_j \in Q'} P(w|Q_j, Q)P(Q_j|Q) \quad (2)$$

Equation 2 shows a query language model mixing the probability of term w given a specific subset of query terms Q_j and Q, weighted by a prior $P(Q_j|Q)$. By assuming $P(w|Q_j, Q) \approx P(w|Q_j)$, we can obtain a simplified version (Equation 3) which has been used in our previous work (Bai et al. 2005):

$$P(w|Q) = \sum_{Q_j \in Q'} P(w|Q_j)P(Q_j|Q) \quad (3)$$

Although Equation 3 provides an effective way to estimate $P(w|Q)$ by combining each $P(w|Q_j)$ (the high-order term association between Q_j and w), it does not take into account the various impacts of different documents or parts of documents that contain w. On the other hand, it does not give a clear description of how to compute the a priori distribution of $P(Q_j|Q)$. In order to solve these issues, we further propose the following extended structure.

2.2 Extended Structure

Our proposed approach aims to integrate query decomposition, document segmentation, and estimation of the a priori distributions into a unified framework. We take into account the following factors:

First, in each relevant document, the actual relevance of different parts of the document may vary. Indeed, there even can be irrelevant information in the document. Therefore, it would seem more reasonable to segment the document into finer-grained parts and use the parts rather than the whole document itself in the query language model derivation process. The detailed methodology for document segmentation will be presented in detail in the next section.

Second, the generation of segments results in another factor, namely the

impact of a segment, which corresponds to the prior probability of a segment in the context of Q_j.

The third factor is the a priori distribution of Q_j. Although a simple assumption of uniform distribution may generate reasonable performance, it is intuitive that each Q_j should have different effect.

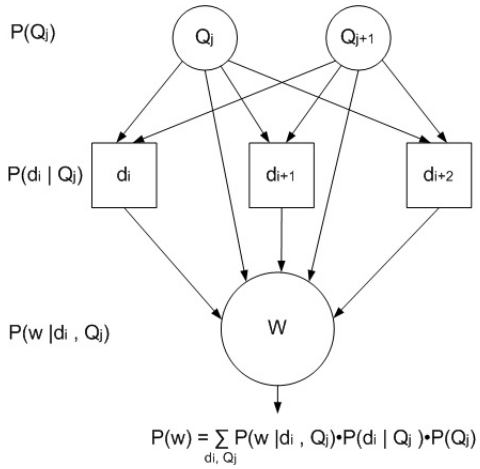


Figure 3: Induction of Structure.

In summary, it is necessary to develop a framework to integrate these considerations. Here, we propose a new graphic model, which is a Bayesian network like structure, as shown in Fig. 3. The directed lines in the figure represent the relations between those objects. Based on the relations among Q_j, d, and w, we extend the Equation 3 by adding in the relations between d_i and Q_j. Then we can obtain a new equation:

$$P(w|\mathbf{Q}) = \sum_{Q_j \in \mathbf{Q}, d \in \mathbf{d}} P(w|Q_j, d)P(d|Q_j)P(Q_j|\mathbf{Q}) \tag{4}$$

where $P(w|Q_j, d)$ represents the probability of the term w being generated given a subset of query terms Q_j and a segment d; $P(d|Q_j)$ is the probability distribution over segments given Q_j; and $P(Q_j|\mathbf{Q})$ is the prior distribution of

query subsets; $\mathbf{d}$ denotes the collection of segments in the feedback documents.

Equation 4 shows a sensible way to estimate $P(w|Q)$. However the conditional probability $P(w|Q_j, d)$ is not easy to estimate. Thus, we simplify Equation 2 by replacing $P(w|Q_j, d)$ with $P(w|Q_j)$. Then a new equation results:

$$P(w|\mathbf{Q}) = \sum_{Q_j \in \mathbf{Q}, d \in \mathbf{d}} P(w|Q_j)P(d|Q_j)P(Q_j|\mathbf{Q}) \tag{5}$$

This allows us to use the aspect model (AM) to estimate the three parameters on the right hand side of the Equation. The details of AM will be presented next.

2.3 Aspect Model

The aspect model is a latent variable model. It associates an unobserved class variable with each observation (Hofmann 1999; Blei and Moreno 2001). Given documents $D \in \mathbf{D} = \{D_1, D_2, \ldots, D_N\}$, and the terms w from a vocabulary $\mathbf{V}$, i.e. $w \in \mathbf{V} = \{w_1, \ldots, w_M\}$, an observation (D, w) is associated with a latent variable $S \in \mathbf{S} = \{S_1, \ldots, S_K\}$. Conceptually, the latent variables are topics embedded in the document collection. One can think of a process where documents generate or "induce" the topics or latent classes, which in turn generate terms according to class specific distributions (Schein et al. 2001). Documents are assumed to be independent of terms, given the topics. The joint probability distribution over documents, topics, and terms is (Schein et al. 2001):

$$P(D, w, S) = P(S)P(D|S)P(w|S) \tag{6}$$

Assuming that S are exhaustive and mutually exclusive, we can sum over the possible values of S when calculating the joint distribution of a document and a term:

$$P(D, w) = \sum_{S} P(S)P(D|S)P(w|S) \tag{7}$$

The parameters in Equation 7 are explained as follows. $P(w|S)$ can be viewed as a language model of latent variable S. $P(D|S)$ is a probability distribution over the training documents. $P(S)$ is the prior distribution on S.

In our application, we prefer using segments of a document rather than the whole document. The motivation is that the different parts (e.g., a sentence, or text within a window) of a document may have different contributions to the aspect model. Let d denote a segment in collection $\mathbf{d} = \{d_1, \ldots, d_N\}$ of pre-segmented documents, w denote a term, and S denote a latent topic.

Given a corpus of N document segments and the words within those segments (w_n^d), the training data for an aspect model is the set of pairs $\{(d_n, w_n^d)\}$ for each segment label and each term in those segments. The Expectation Maximization (EM) algorithm can be used to fit the parameters of Equation 7 from an un-categorized corpus. This corresponds to learning the underlying topics of a corpus $P(w|S)$ as well as the degree to which each training document is about those topics $P(d|S)$ (Blei and Moreno 2001).

In the *E-step*, we compute the posterior probability of the hidden variable given our current model.

E-step:

$$P(S|d, w) = \frac{P(S)P(d|S)P(w|S)}{\sum_{S'} P(S')P(d|S')P(w|S')} \tag{8}$$

In the *M-step*, we maximize the log likelihood of the training data with respect to the parameters $P(S)$, $P(d|S)$ and $P(w|S)$.

M-step:

$$P(d|S) = \frac{\sum_{w \in V} P(S|d,w)n(d,w)}{\sum_{w \in V} \sum_{d' \in \mathbf{d}} P(S|d',w)n(d',w)} \tag{9}$$

$$P(w|S) = \frac{\sum_{d \in \mathbf{d}} P(S|d,w)n(d,w)}{\sum_{w' \in V} \sum_{d \in \mathbf{d}} P(S|d,w')n(d,w')} \tag{10}$$

$$P(S) = \frac{\sum_{d \in \mathbf{d}} \sum_{w \in V} P(S|d,w)n(d,w)}{\sum_{S'} \sum_{w \in V} \sum_{d \in \mathbf{d}} P(S'|d,w)n(d',w)} \tag{11}$$

where $n(d,w)$ is the number of times that the term w appears in the segment d. A detailed discussion can be found in (Hofmann 1999).

It is obvious that the aspect model generates three items, which correspond to $P(w|Q_j)$, $P(d|Q_j)$, and $P(w)$ in Equation 5, respectively. Based on the aspect model, the next subsection will present the procedure of optimization of the model.

3 Model Optimization

In the last section, the algorithms based on two theoretical structures are described. One is a basic structure taking into account query decomposition and the combinations of high-order term relations from different subsets of query. The other is an extended structure by using document segmentation and considering the a priori distribution estimations for both query sets and document segments. In this subsection, we use an Expectation-Maximization (EM) algorithm to optimize the model and use the association rule mining from document segments to set the initial parameter of the model.

3.1 Basic Framework of the Model Optimization

Fig. 4 shows the model optimization process, which consists of five steps: *(1) Pre-segmentation*, *(2) Query decomposition*, *(3) Pre-clustering*, *(4) Parameter Initialization*, and *(5) Model Optimization*. The details are presented as follows.

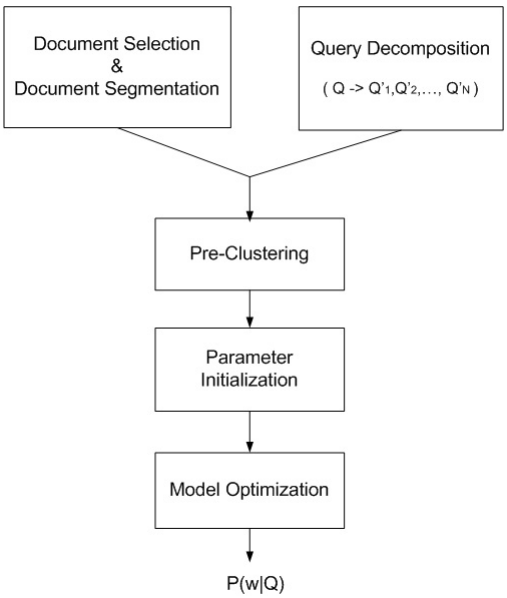


Figure 4: Framework of the model learning and optimization

Pre-segmentation: A number F of (pseudo-) relevance feedback documents is used to derive the query language model. Documents are segmented into chunks using multiple sliding windows of variable lengths. Here, each segmented chunk is treated as a "new" document.

Construction of States: As mentioned in Section 2, the combinations of query terms are considered, i.e. Q_j, as latent variable, i.e. S_{Q_j}. Here, the maximal length of Q_j is set to be three, for reducing the computational cost.

On-the-fly Training Data Construction: In order to build initial training data labelled by the states S_{Q_j}, only those chunks including the query terms in Q_j are selected. These chunks serve as initial training set. The chunks not containing any query terms will be checked for which state they belong to in the next steps.

Parameter Initialization: For estimating the parameters of our model, a simple way is often to initially set the parameters randomly or set them to be the uniform values. For example: $P(S_{Q_j}) = \frac{1}{Num\_of\_subsets\_of\_Q}$. However, in this paper, we will adopt a different way to set the value of $P(w|S_{Q_j})$ by using association rule mining. The reason is due to the fact of data sparsity. Since the number of selected documents (and the segmented chunks) is a relatively small, when using EM algorithm to optimize the model, the over-fitting problem caused by data sparsity may lead the optimization to converge at local maximal point. On the other hand, simply setting the uniform values to those parameters can also lead to over-fitting. The mining of association rules has been proven an effective mechanism for detecting the dependency between item sets in transactional data, in our case, Q_j and the observed word w. Therefore, the initial value of $P(w|S_{Q_j})$ is set based on the association rule $Q_j \Rightarrow w$ derived from the documents where the chunks and terms are considered as transactions and items respectively. The details of mining of association rules from text will be presented in the next subsection. The initial value of $P(d|S_{Q_j})$ is set to be $P(d|S_{Q_j}) = \frac{d}{\sum_i \#d_i}$, where $\#d_i$ is the number of chunk d_i occurring in the chunk collection.

Model Optimization: In the process of optimization, an EM algorithm is adopted. Equations 8 $\sim$ 11 are used to iteratively estimate the parameters, i.e., $P(d|S_{Q_j})$, $P(w|S_{Q_j})$ and $P(S_{Q_j})$, of the model with the clustered chunks.

3.2 Use of Association Rule for Parameter Initialization

Mining association rules is an important technique for discovering meaningful patterns in transaction databases. Formally, the problem can be formulated as follows (Agrawal et al. 1993). Let $I = \{i_1, i_2, \cdots, i_n\}$ be a set of n binary attributes called *items*. Let $D = \{t_1, t_2, \cdots, t_m\}$ be a set of transactions called the *database*. Each transaction in D has a unique transaction ID and contains a subset of the items in I (Hahsler et al. 2008). An association rule is a rule of the form $X \Rightarrow Y$, where $X, Y \subseteq I$, and X, Y are two disjoint sets of items. It means that if all the items in X are found in a transaction then it is likely that the items in Y are also contained in the transaction. The sets of items X and Y are respectively called the *antecedent* and *consequent* of the rule (Hahsler et al. 2008). To select interesting rules from the set of all possible rules, constraints on various measures of significance and strength can be used. The best-known constraints are minimum thresholds on support and confidence.

$$supp(X \Rightarrow Y) = supp(X \cup Y) = \frac{C_{XY}}{M} \tag{12}$$

$$conf(X \Rightarrow Y) = \frac{supp(X \cup Y)}{supp(X)}, \tag{13}$$

where C_{XY} is the number of transactions which contain all the items in X and Y, and M is the number of transactions in the database.

Support, in Equation 12, is defined as the fraction of transactions in the database which contain all items in a specific rule (Agrawal et al. 1993). *Confidence*, in Equation 13, is an estimate of the conditional probability $P(E_Y|E_X)$, where E_X (E_Y) is the event that X (Y) occurs in a transaction (Hipp et al. 2000).

Fig. 5 shows an example of how much dependency could be obtained between

high-combustion fuel create ⇒ *laser* (0.03265, 61.11)
high-combustion fuel hydrogen ⇒ *laser* (0.01929, 100.00)
high-combustion fuel ⇒ *laser* (0.01484, 47.62)
high-combustion fuel energy ⇒ *laser* (0.02671, 56.25)
high-combustion create hydrogen ⇒ *laser* (0.01336, 100.00)

Figure 5: Association Rules

the subsets of query and the observed word "*laser*" by mining the associated rules. In this figure, two values listed on the right-most hand side are the *support* and *confidence* of the associated rules, respectively. Although the *support* of the rule "*high-combustion create hydrogen* ⇒ *laser*" is lower than any others, the two factors *high-combustion* and *hydrogen* have more impacts on the generation of *laser* than the single factor *high-combustion* does. Thus, the confidence of this rule is highest in all the generated rules. According to the definitions of the two measurements in the mining of association rules, *support* simply represents the co-occurrence of query terms and each word over the segments collection, it can not really reflect the implicit relative relationships between them. Unlike *support*, *confidence* is computed as a conditional probability of a word given the query terms, which can represent how good a captured rule is. We therefore believe, in this paper, the confidence of a rule can be used to effectively measure the dependence between the query terms and the observed word occurring in the document.

Accordingly, the *confidence* of association rules are used to compute the conditional probability $P(w|Q_j)$, and then is set to be the initial parameter of aspect model.

$$P(w|Q_j) = \frac{Conf(Q_j \Rightarrow w)}{\sum_{w'} Conf(Q_j \Rightarrow w')} \tag{14}$$

3.3 Query Smoothing

In the construction of theoretical framework, we also notice a phenomenon of query shift when more words are added into a new query language model. This means the impact of the query terms is weaken to some extend when adding more words into the new query model, which easily results in "query shift" and limits the further improvement in document retrieval. Therefore, we further optimize our framework by using a smoothing method, in which the original query model is linearly combined with the expanded query model to reduce the impact of query shift.

In the process of building the original query model, we only consider the distribution of each single query term q_i by computing the product of its term frequency (QTF) in query Q and its inverse document frequency IDF. The IDF of q_i is the logarithm of the number of all documents (document collection used in this paper) divided by the number of documents containing the query term q_i.

Given the original query $Q_o = \{q_1, \cdots, q_{|Q_o|}\}$, the original query model, $P(q_i|Q_o)$ is computed as:

$$P(q_i|Q_o) = \frac{QTF(q_i) * IDF(q_i)}{\sum_{j \in 1 \cdots |Q_o|} QTF(q_j) * IDF(q_j)} \tag{15}$$

To build a new query model $P(w|Q_s)$, the distribution $P(w|Q)$ is then combined with the original query model $P(q_i|Q_o)$ via smoothing, a commonly used technique to combine different models, or term distributions.

Typically, linear mixture, a classical smoothing method, can be used to derive the "new" smoothed model $P(w|Q_s)$:

$$P(w|Q_s) = \lambda P(w|Q) + (1 - \lambda) P(w|Q_o) \tag{16}$$

where $P(w|Q_o) = 0$ when the term w does not occur in the original query.

So far, we have presented our theoretical framework and a detailed description about detecting the high-order contextual term relations. In he following sections, we will present the methodology and results of our extensive empirical evaluation on large scale collections.

4 Data and Experimental Setup

In this section, a description of the data sets used in our experiments and the experimental setup will be given.

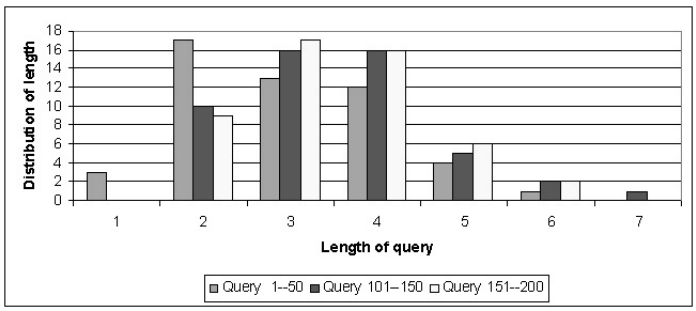
4.1 Data

The experiments are conducted using various TREC [1] collections and query topics shown in Table 1. Four different TREC data sets are used in our experiments. In addition, different fields of the five topic sets are used in different experiments to verify the robustness of our method with respect to different average query lengths.

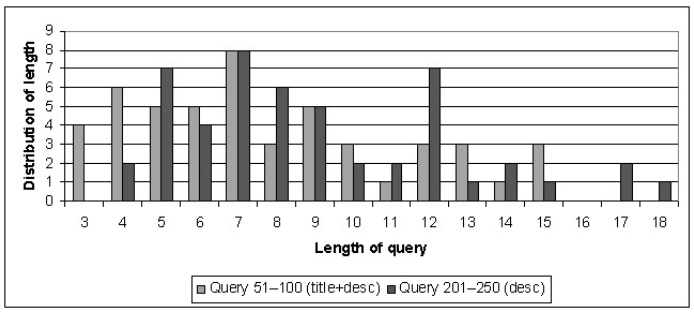
Table 1: Test Collections and Query Topics

Coll.	**Description**	**Size** (MB)	**# Doc.**	**Query**	**Q.fields**
AP89	Associated Press (1989) Disk 1	254	84,678	1–50	title
AP88–89	Associated Press (1988–1989) Disk 1,2	492	164,597	101–150 151–200	title title
WSJ90–92	Associated Press (1990–1992) Disk 2	242	74,520	201–250	desc.
SJM	San Jose Mercury News (1991) Disk3	287	90,257	51–100	title & desc.

[1] TREC (Text Retrieval Conference) is a prominent conference for the evaluation of information retrieval systems. It provides large scale benchmarking collections, test topics and relevance judgments for different information retrieval systems to compare with each other.



(a)



(b)

Figure 6: Distribution of the length of query (a) 1–50, 101–150, 151–200 (only title field) (b) 51–100 (desc+title), 201–250 (desc)

Fig. 6 shows the distribution of the length of queries. Fig. 6(a) is for the queries only with *title* field of topics, and Fig. 6(b) is for the queries with the field of *description* and two fields of *title+description*. All these three types of queries have been widely used in information retrieval experiments. The *title* field simulates typical keyword-based queries, while the *description* (simulating, e.g., a "search by example" scenario) and *title+description* (simulating, e.g., a "search by concept and its description" scenario) both have been used as longer and verbose queries. In particular, the importance of verbose queries has been recognized recently (Kumaran and Allan 2008; Bendersky and Croft 2009). Therefore, in our experiments, we use all the three settings in order to increase the diversity of test queries and to better test the robustness of our model with

respect to query length.

For the use of the *title* field only, the length of queries mostly ranges between 2 and 4. Since the field of *description* gives a more detailed explanation, the length of query combining *title* and/or *description* has a wider distribution in the range of 3∼18, as shown in Fig. 6(b). The average length of the five query sets are respectively 3.2 words for queries 1–50, 3.6 words for queries 101–150, 4.3 words for queries 151–200, 8 words for queries 201–250, and 12.2 words for queries 51–100.

Considering a word perhaps occurring in various inflected forms, the Porter stemmer is used to deal with this case. In addition, a standard stop-word list is used as well to remove those stop words, such as *in, of, to* etc., occurring in the document collection.

4.2 Experimental Setup

In our experiments, the Lemur Toolkit was used to construct the baseline. For association rule mining, the Apriori algorithm implemented in the WEKA toolkit was adapted with the granularity of transactions set to be at the chunk level. As a comparison, we test our methods with different levels of chunks, which are obtained by segmenting documents with *sentence*, *a fixed-length sliding window* and *multiple sliding windows*, respectively. In our experiments, we tested different-size sliding windows, respectively containing from 15 words to 45 words, with 1/5 of the window size being overlapped. The overlapping length is an experimental value. To avoid data sparsity of using one window, we also adopt multiple windows generally including four different-size windows (25, 30, 35, and 40). In addition, we further conduct experiments on the whole documents without segmentation to test the effect of using sliding windows.

We use the top 35 documents as pseudo feedback documents, and the top

100 terms from the new query model are selected. The linear interpolation parameter λ for mixing the original and derived query models is experimentally set to be 0.9. Indeed, our experiments show little variation in performance when λ is more than 0.9.

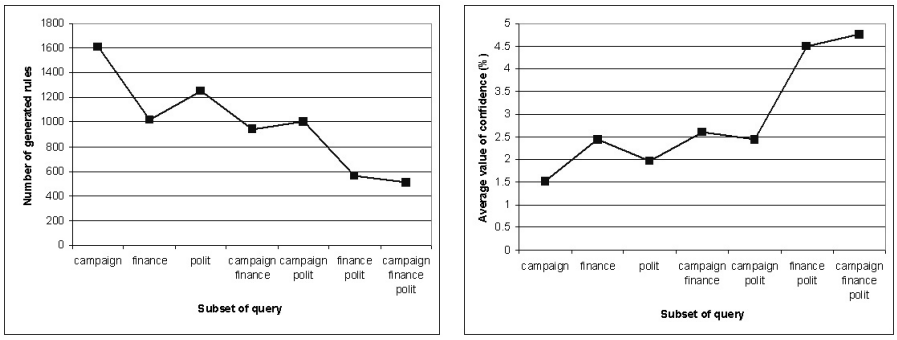
Our proposed Aspect Model (AM) is compared with a baseline language model based on the KL-Divergence (KL), the Relevance Model (RM), and the language model based on Information Flow (IF).

The effectiveness indicators are the standard mean average precision (MAP) and recall, which are calculated based on 1000 retrieved documents for each query. The MAP is computed and averaged across different recall points (i.e., while each relevant document is returned by the system) for each query, and then averaged over all the queries. We also perform the t-test to measure the statistical significance of performance improvement.

5 Empirical Evaluation

5.1 Characteristics of the High-order Context Relations

In Section 3, we proposed to use the *confidence* measure of association rule to estimate the dependency of Q_j and w, and then to compute $P(w|Q_j)$. Here, we further illustrate, by an example, the advantage of query decomposition and the use of association rule mining. Consider the query $Q = \{finance\ campaign\ polit\}$. Fig. 7 shows the number of generated rules and their average values of confidence corresponding to the query and its decomposed subsets. Here, two distinct trends are presented on the decomposed subsets of query. In Fig. 7(a), the longer combination of query terms tends to generate less associated rules, and Fig. 7(b) shows a reverse trend that these longer units have a larger than average *confidence* value of the rules. The phe-

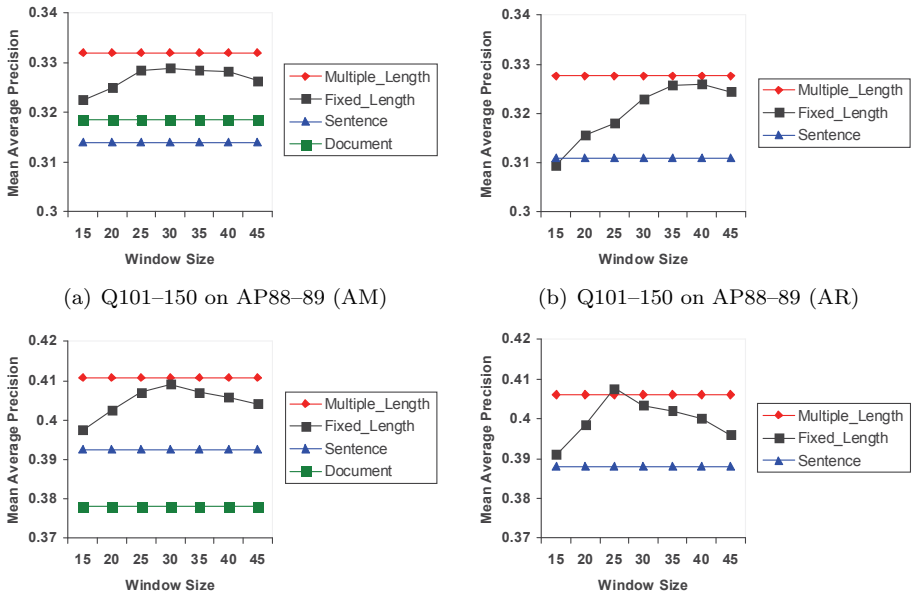


(a) (b)

Figure 7: The number of generated rules and the distribution of their confidence values on an example. Original query (*finance campaign polit*) – Decomposed query ({campaign}, {finance}, {polit}, {campaign finance}, {finance polit}, {campaign finance}, {finance campaign polit})

nomenon in Fig. 7(a) is due to the document segmentation. Hence, the number of words co-occurring with the multiple query terms is generally less than the number of words co-occurring with the single query term. This results in less associated rules being generated. The trend shown in Fig. 7(b) satisfies our intuitive observation that those words often co-occurring with multiple query terms have stronger dependencies on query to some extend. Therefore, it also reflects that query decomposition could have positive impact on the acquisition of the high-order relations for document retrieval.

In order to further describe the impact of query decomposition on the estimation of high-order relations between terms, Table 4 is shown in APPENDIX, in which the 7 decomposed subsets of query {*finance campaign polit*} are listed, and 20 top-ranked words corresponding to each query subset are also shown. We can find these word rankings in various order with different weights given each query subsection. These different word rankings present a fact that the decomposition of query can help us to find some "hidden" information among query terms and words, which can not be found when considering the query



(a) Q101–150 on AP88–89 (AM)

(b) Q101–150 on AP88–89 (AR)

(c) Q151–200 on AP88–89 (AM)

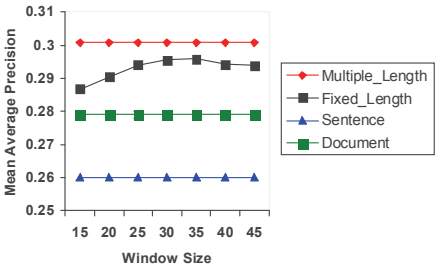
(d) Q151–200 on AP88–89 (AR)

Figure 8: Effects of Multiple Windows (1)

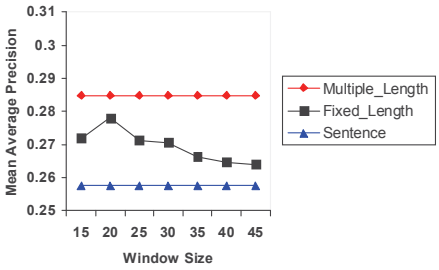
terms individually. Therefore, it is very interesting to see how our method performs by collecting these various context information. In the next section, we will present the related results.

5.2 Result Analysis

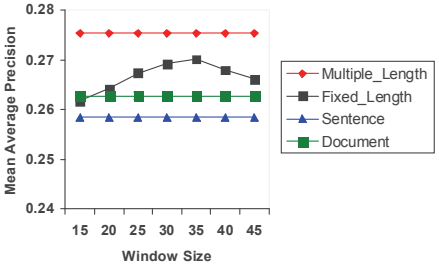
According to our description in the aforementioned sections, some factors can affect the acquisition of the high-order context relations, and hence the performance of document retrieval. Here, we highlight two factors, document segmentation and query decomposition, and the detailed performance data associated with the two factors will be given.



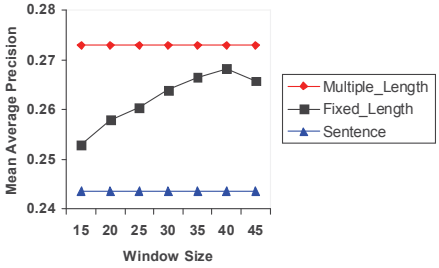
(a) Q201–250 on WSJ9092 (AM)



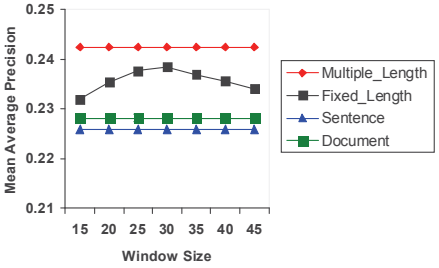
(b) Q201–250 on WSJ9092 (AR)



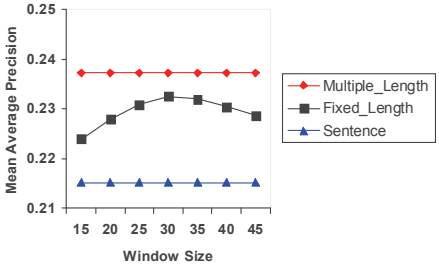
(c) Q1–50 on AP89 (AM)



(d) Q1–50 on AP89 (AR)



(e) Q51–100 on SJM (AM)



(f) Q51–100 on SJM (AR)

Figure 9: Effects of Multiple Windows (2)

5.2.1 The Impact of Document Segmentation

To compare the effectiveness of segmenting the document into chunks, we used three types of window, sentence-based window, a fixed-size sliding window, and multiple fixed-size sliding windows. Fig. 8∼9 show the MAP values of using the three types of window to test five query sets over four TREC data collections, respectively.

According to these figures, it is easy to find that using the multiple-window based segment unit generates the best performance. The performance of using a fixed-length sliding window show a middle-ranked effectiveness, followed by the sentence-based segmentation and the whole document. As we have indicated, the number of sentences that could be extracted from the feedback documents is limited. Therefore, it is not easy to find the co-occurrence of the query terms and the words occurring within the sentences. This probably results in the poor detection and estimation of the high-order context relations due to the data sparsity when using sentence-based unit. The problem gets worse when the whole document is treated as a segment. Therefore the use of an overlapped sliding window, to some extend, can alleviate the impact of data sparsity since the number of segments is much more than the number of sentence. Moreover, the use of the multiple windows generates a better performance than just using a fixed-length window. It is because the multiple windows not only further increase the number of segments, but also consider both the shorter segment units and the longer ones. In general, the longer units can capture more useful relations between terms, and the shorter ones may reduce redundancies. Thus the use of multiple windows can combine these advantages in our system.

5.2.2 Comparison between Two Structures

As a comparison, we also consider only using the basic structure described by Equation 3, in which the value of $P(w|Q_j)$ is derived based on the association rules (AR) only. It is called AR-based method, in contrast with the AM-based method. In Fig. 8∼9, the performance of using various models are presented respectively.

According to the distribution of the obtained MAP values, we can found that using AM can generate better performance than those of using AR-based method over all the data collections. The reasons are as follows:

First, for AR-based method, each segment is used as a transaction, and is considered having equal contribution to the acquisition of the high-order context relations. However, intuitively, some transactions could play more important role than others. Unlike the AR-based method, the AM-based method not only considers the relation of the query term and the observed words in the document ($P(w|Q_j)$), but also the effect of the segments by estimating the relations between the subset of query Q_j and each segment d_i ($P(d_i|Q_j)$).

Second, in AR-based method, the prior distribution of each subset of query $P(Q_j)$ is assumed to be uniform. Although we obtain fair results based on this assumption, we also notice the fact that we often have different focus on each Q_j derived from the query Q. This means simply assigning equal probability distribution to each Q_j probably seems to be crude. Therefore, the AM method is more effective in estimating $P(Q_j)$.

Finally, in comparison with the AR-based method, the AM-based method shows a more stable performance when testing it over five data collections by varying the sizes of the sliding window. Table 2 shows the differences between the two methods. The average MAP and their variance over the 7 different-sized windows when using AM-based method and AR-based method on each

collection are listed respectively. We can find that the variance of the AM performance on every collection is much smaller than that of the AR-based method[2]. It is because the AM-based method takes into account the diverse effects of each segment given the different subset of query Q_j by increasing or reducing the contributions of some segments according to their relevancy to the query. Thus it weakens the variances even if the different-sized sliding windows are used.

	AP89 Q1–50 (title)	AP88–89 Q101–150 (title)	AP88–89 Q151–200 (title)	WSJ90–92 Q201–250 (desc.)	SJM Q51–100 (title+desc.)
AM_Avg.	**0.2665**	**0.3271***	**0.4049**	**0.2927***	**0.2356***
AR_Avg.	0.2622	0.3203	0.3990	0.2698	0.2293
AM_Var.	**0.0023**	**0.0051**	**0.0051**	**0.0025**	**0.0017**
AR_Var.	0.0043	0.021	0.029	0.0040	0.0022

$*$ indicates the difference from AR_Avg is statistically significant at $p-$value < 0.05

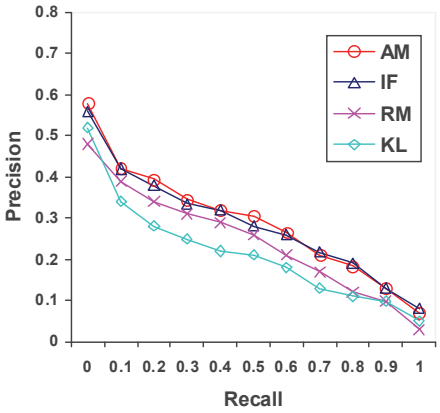
Table 2: Comparison of the average value of MAP over different-sized sliding windows

5.2.3 Comparison between Our Method and Others

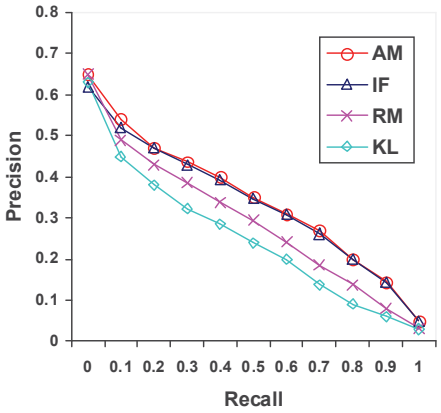
As a further comparison, Table 3 shows the MAPs of AM and three baselines including KL, RM, and IF, where the three columns on the right hand side indicate the MAP improvements of AM over KL, RM, and IF, respectively. In addition, Fig. 10(a) $\sim$ 10(e) also show the Precision/Recall curves of the five approaches.

Tables 3(a), 3(b) and 3(c) show the retrieval performance, which is obtained by running three baselines and our two methods against the AP89 and the AP8889 collections only using the title field of TREC topics. Our approach shows statistically significant improvement over the KL method by about 30% (39.8%, 41.7% and 34%), over the RM by at least 7% (21.3%, 7.9% and 18.3%), and over the information flow model by more than 3% (3.3%, 3.9%, and 4.1%).

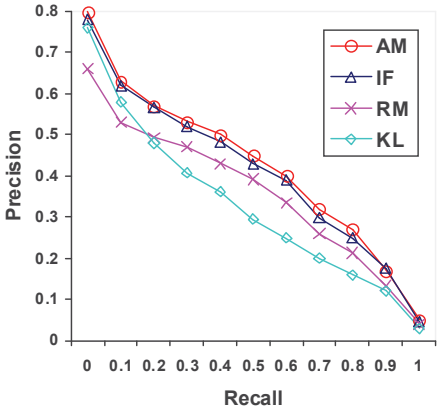
[2]The difference between the average AM_Var and AR_Var (0.0033 vs. 0.012) over all collections passes a significance test at $p < 0.06$.



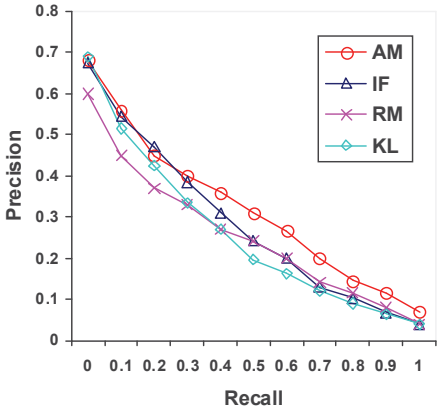
(a) Q1–50 on AP89



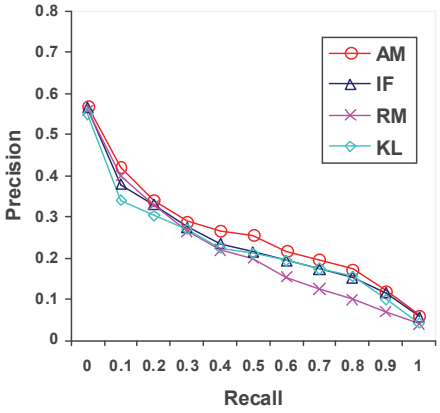
(b) Q101–150 on AP88–89



(c) Q150–200 on AP88–89



(d) Q201–250 on WSJ90–92



(e) Q51–100 on SJM

Figure 10: Precision-recall Curves

Table 3: Comparison between KL, RM, IF

(a) Experimental results on AP89 collection for queries 1–50 (title)

	KL	RM	IF	AM	Impr. (%) over KL	Impr. (%) over RM	Impr. (%) over IF
MAP	0.1970	0.2270	0.2664	0.2754	+39.8**	+21.3**	+3.3
# Relevant Retrieved	1702	2312	2372	2362			

(b) Experimental results on AP88-89 collection for queries 101–150 (title)

	KL	RM	IF	AM	Impr. (%) over KL	Impr. (%) over RM	Impr. (%) over IF
MAP	0.2338	0.3069	0.3185	0.3312	+41.7**	+7.9*	+3.9*
# Relevant Retrieved	3160	3910	3900	3902			

(c) Experimental results on AP88-89 collection for queries 151–200 (title)

	KL	RM	IF	AM	Impr. (%) over KL	Impr. (%) over RM	Impr. (%) over IF
MAP	0.3063	0.3471	0.3942	0.4105	+34**	+18.3**	+4.1*
# Relevant Retrieved	3319	3566	3841	3798			

(d) Experimental results on WSJ90-92 collection for queries 201–250 (description)

	KL	RM	IF	AM	Impr. (%) over KL	Impr. (%) over RM	Impr. (%) over IF
MAP	0.2366	0.2403	0.2673	0.3007	+27.1**	+25.1**	+12.5**
# Relevant Retrieved	978	990	1015	1043			

(e) Experimental results on SJM collection for queries 51–100 (title & description)

	KL	RM	IF	AM	Impr. (%) over KL	Impr. (%) over RM	Impr. (%) over IF
MAP	0.2105	0.2154	0.2201	0.2423	+15.1**	+12.5**	+10.1**
# Relevant Retrieved	1460	1486	1488	1519			

$*$ indicates the difference is statistically significant at the level of $p-$value < 0.05
$**$ indicates the difference is statistically significant at the level of $p-$value < 0.01

Our approach also improves recall over the KL and RM methods. As shown in Figures 10(a), 10(b) and 10(c), our approach achieves better precision than KL and RM at almost all the recall points.

Unlike the above topic sets, the queries are longer when using the field of *description* and the fields of *description* plus *title*, whose average length are 8 and 12.2 words for each query, respectively. It means that longer queries will generate a larger number of subsets of queries in general. Table 3(d) and Fig. 10(d) list the results on the WSJ collection using the description field

of topics 201-250. Our approach also shows significant improvement in MAP over the three baselines by 27.3%, 25.1%, and 12.5%, respectively. Further, the experimental results on the SJM collection are shown in Table 3(e) and Fig. 10(e). Again, significant improvements (15.1%, 12.5%, and 10.1%) in term of MAP have been achieved.

The performance improvements of the three query expansion models over the KL baseline on longer queries are not as much as the improvements obtained by using shorter queries. This is due to the query length. In general, the longer the query is, the more useful information it may have contained. Thus, it is reasonable to find that, for longer queries, even the baselines could achieve good performance. However, even in this case, our approach still shows its advantage in capturing the relationships between query terms and words in pseudo feedback documents.

Among the three query expansion models, both IF and AM largely outperform the RM on all the collections. Particularly for longer queries, the RM's MAPs are more or less similar to those of the KL baseline. Therefore, the IF and AM are less sensitive to the query length. In addition, the AM has shown significant improvements (up to 12.5%) over the IF, which is already a strong query expansion model, in the experiments. Moreover, more improvements are obtained when the queries are longer. This reflects the effects of query decomposition, i.e., the consideration of the contributions from any parts of the query and the document segmentation will improve retrieval effectiveness. It is also a good demonstration of the robustness of the AM approach with respect to query length.

It is worth noting that the majority of improvement of the AM method over the baselines comes from the model itself rather than the contribution from document segmentation. As shown on the right side of each graph in Fig. 8∼9,

using the whole document as a segment for the AM method generally degrades the performance by about 2∼5% only when compared with the best performance using sliding windows. However, the AM method can still generate much better performance than the baselines (shown in Table 3).

5.3 Discussions

So far, we have analyzed the performance of using various techniques for detecting and estimating the high-order context relations from corpora In this paper, we formulate high-order context relations as conditional probabilities between the query and the observed words occurring in the selected documents based on the following novel methods:

Remark (1): The query decomposition makes it possible to not only consider the query as a whole or consider each query term independently, but also combine all the relations rooted on the different subsets (aspects) of the query. Such decomposition expands the query observation space. So our approach has shown significant improvement over the RM and the IF methods.

Remark (2): In the process of dealing with the selected documents, the application of sliding windows helps us focus more on the highly relevant segments rather than the whole document itself. The application of multiple sliding window reduces the possible disadvantage from the use of a single sliding window. The effect of using multiple sliding windows has been shown by the experimental results (Fig. 8∼ 9).

Remark (3): The application of the mining of association rule shows a way to detect the high-order context relations by considering each segment as a transaction and using the measure of *Confidence* to estimate the relations. Further, by considering the subsets of query as "hidden" states in the Aspect Model, the important parameters such as $P(Q_j)$, $P(d|Q_j)$ and $P(w|Q_j)$ can be

optimized through an EM algorithm. The impact of parameter optimization and association rule mining on retrieval effectiveness has been shown in Table 2.

Remark (4): In addition, we measured the efficiency of the proposed AM method vs the direct use of the information flow (IF) model, by recording the elapsed time of the query expansion process. All the experimental runs were based on the configuration of a computational server with 1333 MHz CPU and 2 GB main memory. Our methods were implemented in Perl. For the association rule mining in the AM method, the Perl program calls the Apriori algorithm implemented in the WEKA toolkit. The average elapsed time to complete the query expansion process is approximately 0.9 second (for AM) and 0.7 second (for IF) per query. This efficiency gap is mainly due to that the AM takes into account the association rules mined from different the query subsets and also involves the on-the-fly EM optimization, while the IF is computed from the whole set of query terms only. However, we consider it as a worthwhile tradeoff. The small gap of 0.2 seconds in elapsed time, which can possibly be further reduced through a more efficient implementation of the model, e.g., in C, is indeed not much noticeable to users, but it does lead to significant improvements, up to 12.5%, in effectiveness (MAP), over the IF.

6 Conclusions and Future Work

We have proposed a novel Aspect query language modelling approach based on the acquisition and optimization of high-order contextual term associations. For a query, we decompose it into the multiple subsets instead of seeing it as a whole or treating individual query terms independently. Each relevance feedback document is segmented into variable sized chunks by using multiple sliding windows. Then the estimation of the contextual term associations is done by

utilizing an Aspect Model based method, in which our framework takes into account automatically derived multiple levels of "higher-order" term associations through association rule mining from the document chunks. A series of rigorous experiments have been conducted based on various TREC collections; our approach outperforms a baseline language model and two state-of-the-art query language models, namely the Relevance Model and the Information Flow model. Based on the experimental results, we can draw the following conclusions:

- The method proposed in this paper considers the contributions from the different combinations of query words, i.e., the Q_j in Equation 3. This demonstrates that the incorporation of query decomposition which takes into account all possible inferences from the query is beneficial to retrieval performance.

- The multiple length document segmentation with overlapping sliding windows contributes to alleviate the problem of data sparseness and hence facilitate the discovery of useful high-order term associations.

- The use of an aspect model combined with association rule mining provides an effective way to estimate the high-order terms relationships. Our method has proved more effective than the Information Flow model. This is, in our opinion, a significant step forward for developing operational query language models.

In the future, we will further our work in the following directions. First, the a priori knowledge of feedback documents, such as the likelihood obtained after initial retrieval, could be considered to improve the effectiveness of our current system. Second, the distribution of the position of the words around query terms may also be an interesting factor to explore. Third, the introduction of additional linguistic information may be useful to discover more "hidden" rela-

tionships which cannot be found by only using statistical approaches. Fourth, other types of term relationships such as those from the WordNet, will be incorporated. Finally, further experiments will be conducted on more TREC collections and our approach will be compared with other state-of-the-art algorithms such as Markov Random Fields (MRF).

7 Acknowledgments

This work is funded in part by the UK's Engineering and Physical Sciences Research Council (grant number: EP/F014708/1). We thank the anonymous reviewers for their constructive comments and Peng Zhang for his assistance in collecting and analyzing some of the experimental results.

References

R. Agrawal, T. Imielinski, and A. Swami. 1993. Mining association rules between sets of items in large databases. pages 207–216.

J. Bai, D. Song, P. Bruza, J.-Y. Nie, and G. Cao. 2005. Query expansion using term relationships language models for information retrieval. In *Proceedings of the 14th Conference on Information and Knowledge Management (CIKM'2005)*, pages 688–695, Bremen, Germany.

D. M. Blei and P. J. Moreno. 2001. Topic segmentation with an aspect hidden markov model. In *Proceedings of the 24th Annual ACM Conference of Research and Development in Information Retrieval (SIGIR'2001)*, pages 343–348, New Orleans, Louisiana, USA.

M. Bendersky and B. Croft. 2009. Discovering key concepts in verbose queries. In *Proceedings of the 31st Annual International ACM SIGIR Conference on*

Research and Development in Information Retrieval(SIGIR'2009), pages 491–498, Singapore.

G. Cao, J. Nie, and J. Bai. 2005. Integrating term relationships into language models. In *Proceedings of the 28th Annual ACM Conference of Research and Development in Information Retrieval (SIGIR'2005)*, pages 298–305.

C. Creighton and S. Hanash. 2003. Mining gene expression databases for association rules. *Bioinformatics*, 19(1):79–86.

J. Gao, J. Nie, G. Wu, and G. Cao. 2004. Dependence language model for information retrieval. In *Proceedings of the 27th Annual ACM Conference of Research and Development in Information Retrieval (SIGIR'2004)*, pages 170–177.

G. Kumaran and J. Allan. 2008. effective and Efficient User Interaction for Long Queries. In *Proceedings of the 31st Annual ACM Conference of Research and Development in Information Retrieval (SIGIR'2008)*, pages 11–18.

M. Hahsler, C. Buchta, and K. Hornik. 2008. Selective association rule generation. *Computational Statistics*, 23:303–315.

J. Hipp, G. U., and N. G. 2000. Algorithms for association rule mining - a general survey and comparison. *ACM SIGKDD Explorations NewsLetter*, pages 58–64.

T. Hofmann. 1999. Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual ACM Conference of Research and Development in Information Retrieval (SIGIR'1999)*, pages 50–57, Berkeley, CA, USA.

P. Ingwersen. 2001. Users in context. In *Lectures on Information Retrieval (LNCS 1980)*, pages 157–178.

P. Ingwersen and N. Belkin. 2004. Information retrieval in context (irix). *ACM SIGIR Forum*, 38(2):50–52.

P. Ingwersen and K. Jarvelin. 2005. *The Turn: Integration of Information Seeking and Retrieval in Context*. Springer/Kluwer.

G. J. F. Jones and P. J. Brown. 2004. The role of context in information retrieval. In *In SIGIR'2004 Workshop on Information Retrieval in context (IRiX)*.

J. Lafferty and C. Zhai. 2001. Document Language Models, Query Models, and Risk Minimization for Information Retrieval. In *Proceedings of the 24th Annual ACM Conference of Research and Development in Information Retrieval (SIGIR'2001)*, pages 111–119. ACM Press.

R. Lau, P. Bruza, and D. Song. 2008. Towards a Belief Revision Based Adaptive and Context Sensitive Information Retrieval System. *ACM Transactions on Information Systems*, 26(2).

V. Lavrenko and W. B. Croft. 2001. Relevance-based language models. In *Proceedings of the 24th Annual ACM Conference of Research and Development in Information Retrieval (SIGIR'2001)*, pages 120–127, New York.

J. Luo and S. Bridges. 2000. Mining fuzzy association rules and fuzzy frequency episodes for intrusion detection. *International Journal of Intelligent Systems*, 15(8):687–703.

D. Metzler and W. B. Croft. 2007. Latent concept expansion using markov random fields. In *Proceedings of the 30th Annual ACM Conference of Research and Development in Information Retrieval (SIGIR'2007)*, pages 311–318.

J. Pickens and A. MacFarlane. 2006. Term context models for information re-

trieval. In *Proceedings of the 15th Conference on Information and Knowledge Management (CIKM'2006)*, pages 559–566.

J. Ponte and W. Croft. 1998. A language modeling approach to information retrieval. In *Proceedings of the 21st Annual ACM Conference of Research and Development in Information Retrieval (SIGIR'1998)*, pages 275–281.

A. I. Schein, A. Popescul, and L. H. Ungar. 2001. Pennaspect: A two-way aspect model implementation. Technical Report MS-CIS-01-25, University of Pennsylvania.

X. Shen, B. Tan, and Z. C. 2005. Context-sensitive information retrieval using implicit feedback. In *Proceedings of the 28th Annual ACM Conference of Research and Development in Information Retrieval (SIGIR'2005)*, pages 43–50.

D. Song and P. D. Bruza. 2003. Towards context sensitive information inference. *Journal of the American Society for Information Science and Tecnology*, 54 (3):321–334.

J. Srivastava, R. Cooley, M. Deshpande, and P.-N. Tan. 2000. Web usage mining: Discovery and applications of usage patterns from web data. *SIGKDD Explorations*, 1(2):12–23.

C. Zhai and J. Lafferty. 2001. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th Annual ACM Conference of Research and Development in Information Retrieval (SIGIR'2001)*, pages 334–342.

APPENDIX

	campaign	**finance**	**polit**	**campaign finance**	**campaign polit**	**finance polit**	**campaign finance polit**
1	campaign 0.0496	campaign 0.0763	campaign 0.0646	campaign 0.0447	polit 0.0448	campaign 0.068	polit 0.0428
2	polit 0.0435	financ 0.0423	polit 0.0463	financ0.0447	campaign 0.0448	polit 0.0394	finance 0.0428
3	financ 0.0432	polit 0.0297	financ 0.0257	polit 0.0301	financ 0.0304	financ 0.0394	campaign 0.0428
4	bush 0.0185	bush 0.0165	committe 0.0209	bush 0.0177	committe 0.0212	bush 0.0208	bush 0.0221
5	elect 0.0136	public 0.0144	bush 0.0188	public 0.0141	bush 0.0187	contribut 0.0166	committe 0.0186
6	contribut 0.013	reform 0.0118	contribut 0.0180	reform 0.0131	action 0.018	committe 0.0155	contribut 0.0176
7	committe 0.0127	congression 0.0112	action 0.0179	congression 0.0125	contribut 0.0165	public 0.0154	action 0.0176
8	candid 0.0125	democrat 0.0106	candid 0.0144	democrat 0.0112	monei 0.0134	action 0.0147	public 0.0151
9	pac 0.0124	congress 0.0106	monei 0.0141	spend 0.011	candid 0.0131	congression 0.0125	congression 0.0145
10	monei 0.0122	limit 0.0106	pac 0.0137	congress 0.011	law 0.0113	candid 0.0113	candid 0.0124
11	senat 0.0115	spend 0.0105	parti 0.0108	limit 0.0108	elect 0.0111	limit 0.0108	propos 0.0114
12	democrat 0.0113	contribute 0.0103	law 0.0107	candid 0.0106	fund 0.0093	monei 0.0107	law 0.011
13	reform 0.0106	candid 0.01	elect 0.0105	contribut 0.0106	propos 0.0092	propos 0.099	limit 0.0103
14	law 0.0105	republican 0.0095	propos 0.0097	law 0.0099	parti 0.0089	democrat 0.0098	reform 0.01
15	presid 0.0097	elect 0.0092	democrat 0.0095	elect 0.0097	feder 0.0087	spend 0.0098	pac 0.01
16	republican 0.0097	law 0.0092	limit 0.0092	presid 0.0096	interest 0.0083	law 0.0097	spend 0.0096
17	spend 0.0095	monei 0.0091	fund 0.0088	republican 0.0094	democrat 0.0083	pac 0.0091	feder 0.0093
18	congress 0.0094	presid 0.0089	republican 0.0086	pac 0.0093	presid 0.0079	republican 0.0089	special 0.0079
19	public 0.0093	pac 0.0087	interest 0.0083	monei 0.0092	public 0.0079	congress 0.0085	interest 0.0072
20	hous 0.0092	propos 0.0078	feder 0.0080	year 0.0085	limit 0.0077	reform 0.0082	year 0.0072

Table 4: Probability of the top-ranked 20 words given each subset of a query. Original query (*finance campaign polit*) – Decomposed query ({campaign}, {finance}, {polit}, {campaign finance}, {finance polit}, {campaign finance}, {finance campaign polit})

List of Figures

1 Example of the query decomposition 8

2 A graphic model of the relations between the subset of query Q_j and the observed word w_t 8

3 Induction of Structure. 10

4 Framework of the model learning and optimization 14

5 Association Rules 17

6 Distribution of the length of query (a) 1–50, 101–150, 151–200 (only title field) (b) 51–100 (desc+title), 201–250 (desc) 20

7 The number of generated rules and the distribution of their confidence values on an example. Original query (*finance campaign polit*) – Decomposed query ({campaign}, {finance}, {polit}, {campaign finance}, {finance polit}, {campaign finance}, {finance campaign polit}) 23

8 Effects of Multiple Windows (1) 24

9 Effects of Multiple Windows (2) 25

10 Precision-recall Curves 29

List of Tables

1 Test Collections and Query Topics 19

2 Comparison of the average value of MAP over different-sized sliding windows 28

3 Comparison between KL, RM, IF 30

4 Probability of the top-ranked 20 words given each subset of a query. Original query (*finance campaign polit*) – Decomposed query ({campaign}, {finance}, {polit}, {campaign finance}, {finance polit}, {campaign finance}, {finance campaign polit}) 39